Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.070866

ARTICLE

Optimizing Resource Allocation in Blockchain Networks Using Neural Genetic Algorithm

Malvinder Singh Bali[1], Weiwei Jiang[2,*], Saurav Verma[3], Kanwalpreet Kour[4] and Ashwini Rao[3]

[1]Department of Data Science & Engineering, Manipal University Jaipur, Jaipur, 303007, India
[2]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[3]SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering, Mumbai, 400056, India
[4]School of Computer Science & Engineering, Manipal University Jaipur, Jaipur, 303007, India

*Corresponding Author: Weiwei Jiang. Email: jww@bupt.edu.cn

Received: 25 July 2025; Accepted: 01 October 2025; Published: 09 December 2025

ABSTRACT: In recent years, Blockchain Technology has become a paradigm shift, providing Transparent, Secure, and Decentralized platforms for diverse applications, ranging from Cryptocurrency to supply chain management. Nevertheless, the optimization of blockchain networks remains a critical challenge due to persistent issues such as latency, scalability, and energy consumption. This study proposes an innovative approach to Blockchain network optimization, drawing inspiration from principles of biological evolution and natural selection through evolutionary algorithms. Specifically, we explore the application of genetic algorithms, particle swarm optimization, and related evolutionary techniques to enhance the performance of blockchain networks. The proposed methodologies aim to optimize consensus mechanisms, improve transaction throughput, and reduce resource consumption. Through extensive simulations and real-world experiments, our findings demonstrate significant improvements in network efficiency, scalability, and stability. This research offers a thorough analysis of existing optimization techniques, introduces novel strategies, and assesses their efficacy based on empirical outputs.

KEYWORDS: Blockchain technology; energy efficiency; environmental impact; evolutionary algorithms; optimization

1 Introduction

1.1 Background and Motivation

Blockchain technology has transformed several sectors by providing a decentralized, transparent, and immutable platform for recording transactions [1]. As a distributed ledger technology, blockchain employs cryptographic hashes and consensus mechanisms to ensure security and transparency. Despite its numerous advantages, blockchain faces significant issues, regarding scalability, latency, and energy consumption [2]. Energy optimization is a critical concern in blockchain technology, because it has a direct effect on system efficiency, financial viability, and the sustainability of the environment. [3]. Prominent blockchains like Bitcoin and Ethereum (before switching to Proof of Stake) use the consensus mechanism called Proof of Work (PoW), which requires a substantial processing power [4]. This results in exorbitant energy consumption. For example, every year, the energy consumption of bitcoin mining alone surpasses that of certain entire nations [5]. As of 2021, the Cambridge Centre for Alternative Finance predicted that Bitcoin's yearly energy consumption was 138 TWh [6]. Moreover, PoW mining frequently uses energy, which raises carbon emissions significantly. As per the study of Nature Communications, the annual carbon footprint of Bitcoin mining is between 22 and 29 MtCO2, which is equivalent to the carbon footprint of large towns.

Additionally, excessive energy use raises miners' operating expenses, which might make mining unprofitable, especially when cryptocurrency values are down. Network security and decentralization may be jeopardized if participation is discouraged by this fluctuation in income. Additionally, the concentration of mining power among entities with access to inexpensive electricity and the decentralized nature of blockchain is compromised by powerful tools, which also raises the possibility of 51% incidents, in which one party controls most of the network's mining capacity.

Blockchain networks can increase decentralization, encourage wider participation, and fortify network security by reducing energy consumption. Energy optimization in blockchain is not only essential for the environment but also for the financial viability and security of blockchain networks. The blockchain sector may lessen its environmental effect while preserving strong, safe, and decentralized networks by implementing energy-efficient consensus procedures, increasing hardware efficiency, and developing innovative energy management techniques.

In this paper, we tackle the problem of minimizing energy usage in blockchain networks by using Bio-Inspired Evolutionary Algorithms (EAs). These algorithms optimize complicated problems by utilizing concepts like crossover, mutation, and selection that are derived from natural evolution. Our research is structured in three phases: First, we identify key areas for Blockchain Optimization; second, we investigate various evolutionary algorithms applicable to these optimization challenges and finally, we implement an evolutionary algorithm to optimize energy consumption and reduce carbon emissions associated with blockchain technology.

1.2 Contribution

1. To Investigate different optimization strategies that are relevant to blockchain technology.
2. To find the best evolutionary algorithm for improving blockchain functionality.
3. To optimize important aspects of blockchain networks using evolutionary algorithms, with an emphasis on lowering carbon emissions and energy usage.

This paper's remaining sections are arranged as follows: An introduction to evolutionary algorithms and blockchain technology is given in Section 1. The many optimization techniques used in blockchain technology are examined in Section 2. Section 3 reviews prior research on blockchain optimization, along with the problem definition and objectives. Section 4 discusses the optimization models designed to achieve the stated objectives. Section 5 outlines the experimental scenarios and evaluation metrics used in this study. Section 6 presents the simulation results conducted in MATLAB, accompanied by a detailed discussion. Finally, Section 7 concludes the paper and highlights potential future research directions.

2 Blockchain and Evolutionary Algorithms

2.1 Blockchain Technology

This technology records transactions across a dispersed network of computers to create a digital ledger that is decentralized that guarantees security, immutability, and transparency. Each record contains a list of transactions, called a "block," which is cryptographically linked to the block before it to create a "chain." The structure ensures that no block can be altered or removed without affecting all subsequent blocks, requiring network-wide consensus for any changes. Key characteristics of blockchain technology include:

1. **Decentralization:** Each device in the blockchain network maintains a comparable copy of the ledger ensuring that conventional method of maintaining Central authority is removed.

2. **Transparency:** Each interconnect node can see transactions on a blockchain, and this helps to promote accountability and limits the risk of fraud.
3. **Security:** Here, Blockchain uses cryptographic methods to protect transaction information. Every block contains a cryptographic hash of the previous block, ensuring the integrity and immutability of the chain.
4. **Consensus Mechanisms:** Consensus methods like Proof of Work (PoW) and Proof of Stake (PoS) are used by blockchain networks to verify transactions and reach consensus among users.

Beyond serving as the foundation for cryptocurrencies like Bitcoin and Ethereum, applications for blockchain technology are numerous and span a variety of sectors. It provides a robust and innovative framework for secure, transparent, and efficient data management.

2.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) [7,8] offer a promising approach to addressing these challenges. EAs are optimization techniques inspired by the principles of genetic evolution and natural selection, capable of exploring vast and complex search spaces to look for near-optimal solutions. This study explores the application of evolutionary based algorithms to optimize blockchain networks, with a focus on enhancing transaction throughput, resource utilization, and consensus mechanisms. Evolutionary algorithms belong to a class of optimization techniques that emulate biological evolution. A binary algorithm, which integrates concepts from genetic algorithms, is employed to create a robust optimization and learning system. Various Bio-inspired Genetic Algorithms are shown in Fig. 1. Genetic Algorithms (GA) [9] and Binary Genetic Algorithms (BGA) are combined with the process of genetic evolution principles to binary-encoded solutions for optimization problems.

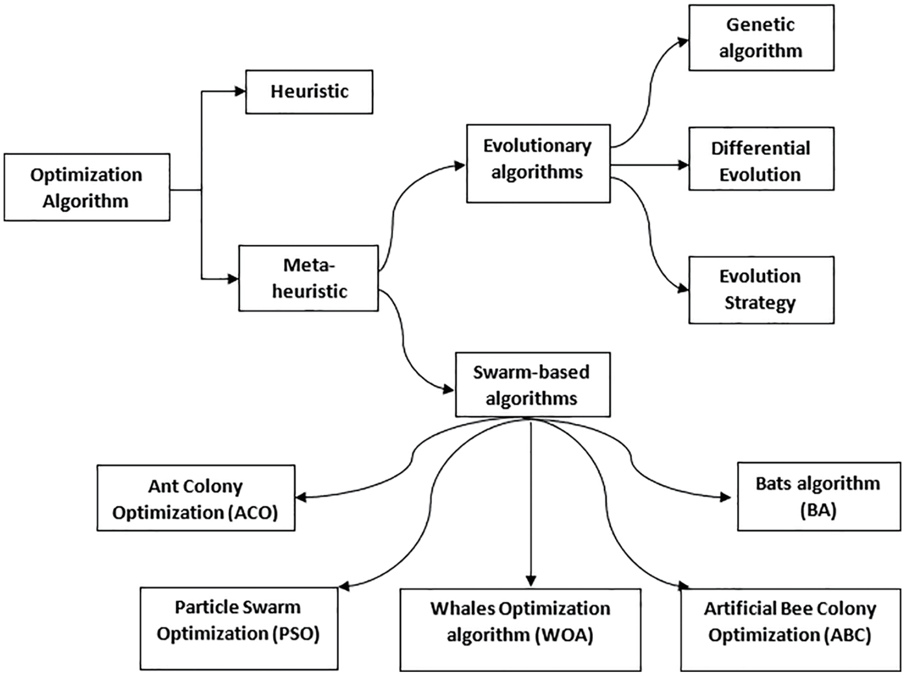

Figure 1: Various bio-inspired optimization methods

The process can be summarized as follows:

First, *encoding* transforms potential solutions into binary sequences (such as 1100101), a format ideal for variables that align naturally with binary representation. Next, an *initial population* is created by generating random binary strings, forming a starting pool of candidate answers. Each of these strings then undergoes *fitness evaluation*, where a performance metric determines their effectiveness in addressing the problem. Following this, *selection* prioritizes higher-performing binary sequences for replication, often using techniques like tournament selection or roulette wheel sampling to identify promising candidates.

During *crossover*, selected pairs exchange segments of their binary codes at designated points, mimicking genetic recombination to produce novel offspring and diversify the population. Subsequently, *mutation* introduces randomness by altering individual bits within these offspring with low probability, fostering exploration of untested regions in the solution landscape. The *replacement* phase then integrates new offspring into the existing group, either partially or entirely, to maintain equilibrium between old and new solutions—strategies here may involve generational turnover or incremental updates.

Finally, the cycle of selection, crossover, mutation, and replacement repeats iteratively until *termination criteria*—such as meeting a target fitness threshold or exhausting a set number of cycles—are fulfilled. This iterative approach balances exploitation of high-quality solutions with exploration of new possibilities, optimizing outcomes over successive generations. This integrated technique is well-suited for a range of combinatorial and discrete optimization issues because it efficiently seeks and optimizes solutions in discrete binary spaces by fusing the concepts of evolutionary algorithms with binary encoding.

2.3 Neural Genetic Algorithm

The Neural Genetic Algorithm (NGA) combines the exploration capabilities of Genetic Algorithms (GA) with the learning power of Neural Networks (NN) to achieve adaptive optimization. In this architecture, the GA initializes a population of candidate solutions, which are fed into a neural network model for performance evaluation and prediction of fitness. The neural network, trained iteratively with evolving populations, refines its predictive accuracy to guide GA operations such as selection, crossover, and mutation. This hybrid interaction accelerates convergence by reducing redundant evaluations and focusing search efforts on high-potential solution spaces [10]. The GA provides global search diversity, while the NN captures underlying patterns in the search landscape, enabling faster adaptation to complex and dynamic environments like blockchain optimization [11]. The training process involves backpropagation to fine-tune weights using historical solution-fitness pairs generated during the GA cycle. This cooperative loop between GA and NN enhances solution quality, scalability, and computational efficiency, making NGA particularly effective for energy optimization, consensus tuning, and transaction scheduling in blockchain networks.

3 Literature Review

3.1 Previous Work

Zheng et al. (2020) explored the application of Genetic Algorithms (GAs) to optimize blockchain parameters, aiming to reduce latency and enhance transaction throughput [12]. This research [13] addresses the deployment of blockchain networks in mobile environments, focusing on the computational challenges associated with Proof-of-Work (PoW) puzzle solving. Most mobile devices lack the computational capacity to support PoW, which is critical for maintaining ledger security and consensus, particularly in networks like Bitcoin. The problem is formulated as a complex nonlinear integer programming problem.

To address the resource allocation challenge, a genetic algorithm-based approach is proposed. Numerical simulations demonstrate that the proposed method converges rapidly to an optimal solution, exhibiting low time complexity and making it suitable for practical applications.

The study in [14] proposes a blockchain-based strategy for optimizing supply chain performance under environmental constraints, offering benefits such as reduced operational costs and lower carbon emissions. By integrating genetic algorithms, the approach provides a robust solution to complex production allocation problems in multi-echelon supply chains (MESC). The findings assist supply chain managers and regulators in making informed decisions, thereby promoting sustainable supply chain practices. In [15], an enhanced Two_Arch2 algorithm is introduced to address scalability and performance challenges in blockchain-enabled Industrial Internet of Things (IIoT) systems. The proposed many-objective optimization model and improved algorithm offer a comprehensive solution to critical issues such as scalability, decentralization, latency, and cost in blockchain-integrated IIoT environments.

A secure, immutable, and autonomous data management system based on blockchain is proposed in [16] to address patient privacy concerns in healthcare. The integration of Discrete Wavelet Transform (DWT) enhances security, while a genetic algorithm optimizes the queuing system. The system incorporates comprehensive logging and a cryptographic key generator to ensure secure and accountable data access. Performance evaluations demonstrate the system's scalability and efficiency, making it a viable solution for safeguarding patient data in medical settings. According to [17], the effective integration of blockchain technology in the Internet of Things (IoT) requires addressing scalability challenges, particularly those related to inefficient peer-to-peer communication. While existing solutions focus on consensus algorithms and storage requirements, improving network topologies and communication protocols is essential for achieving scalable blockchain integration in IoT systems.

Paper in [18] introduces Proof of Federated Training (PoFT), a novel architectural framework for cross-chain verifiable model training using federated learning. Unlike existing blockchain-based FL systems that rely on structural embedding of models, PoFT embeds models via their parameters, enabling efficient and secure cross-chain model exchange and verification. It is the first approach to support federated training across multiple independent blockchain networks. Large-scale experiments on Amazon EC2 demonstrate that PoFT improves model efficacy significantly while incurring only marginal overhead during inter-chain exchanges.

This paper introduces an intelligent system that integrates blockchain technologies, 5G edge networks, and federated learning (FL) to create an efficient and secure framework for collaborative machine learning across multiple organizations. The proposed approach addresses challenges such as data privacy, trust, and interoperability in cross-silo environments. By leveraging blockchain's decentralized nature and FL's privacy-preserving capabilities, the system enables secure model training without the need to share raw data. The authors demonstrate the feasibility and effectiveness of their approach through simulations and case studies, highlighting its potential for applications in 5G and beyond. [19].

This paper [20] proposes a blockchain-based trust management framework for cloud computing environments. It addresses the challenges of ensuring trustworthiness in cloud services by leveraging blockchain's decentralized and immutable ledger to record and verify service interactions. The proposed system enhances transparency and accountability in cloud service provisioning.

This paper proposes a blockchain-based federated learning system with a reputation and incentive mechanism to help home appliance manufacturers build smart home systems while preserving customer privacy. By integrating differential privacy, a novel normalization technique, and decentralized model aggregation, the system ensures security, traceability, and improved prediction accuracy [21]. The summary of the existing work is shown in Table 1.

Table 1: A summary of literature

Aspect	Current work	Previous work			
		BlockSC (Spatial Crowdsourcing in Metaverse) [22]	**Federated cross-chain FL [23]**	**BC security framework in (SDN) [24]**	**Proof of learning (PoLe) [25]**
Scope & goal	Optimizes energy, carbon, OCT, decentralization, and reputation using NN + GA hybrid for blockchain miners and consensus nodes.	Focus on privacy-preserving spatial crowdsourcing with decentralized task assignment and proof generation.	Ensures secure, privacy-aware federated learning across multiple chains for smart grids.	It targets common threats in distributed SDN environments.	Redefines consensus by replacing brute-force PoW with useful ML training.
Core technical approach	Hybrid GA for search + NN surrogate for faster convergence; multi-objective optimization with penalties for constraint violations.	Cryptographic proofs + on-chain contracts for privacy-preserving task allocation; non-ML approach.	Cross-chain protocols, ZK proofs, and FL aggregation with privacy enhancements.	Implement a private Ethereum blockchain to manage authentication, access control.	Consensus redesign where nodes perform ML training; uses a secure validation layer.
Privacy, trust & data exposure	Works with aggregated metrics; avoids exposure of raw private data; NN trained on historical/synthetic data.	Guarantees location privacy and secure task verification.	Privacy-first; no raw data sharing; integrates DP and blockchain anchors.	Focuses on trust and topology privacy but not deep privacy safeguards for user data.	Requires sharing of training/testing data; privacy risks unless mitigated.
Decentralization, overhead & scalability	Decentralized GA; NN surrogate reduces costly evaluations; potential centralized overhead if NN training isn't distributed.	Semi-decentralized; scalability challenged by blockchain transaction costs.	High coordination overhead; suitable for permissioned or hybrid environments.	Trades pure decentralization for latency and reliability; semi-centralized.	Verifying ML and distributing datasets adds high overhead and impacts scalability.
Evaluation metrics & results	Energy, Carbon, OCT, Decentralization, Reputation (RT); shows significant energy/OCT reduction vs. GA/PSO baselines.	Privacy guarantees, task accuracy, verification latency.	Model accuracy, convergence speed, privacy leakage, communication overhead.	Trust coordination latency, task orchestration efficiency.	Energy efficiency, block generation stability, and ML utility.
Strengths	Multi-objective optimization; faster convergence; explicit trade-off handling; quantified gains vs. baselines.	Strong privacy; provable guarantees; decentralized trust in metaverse applications.	High security; privacy-preserving training; cross-chain trust.	Domain-optimized trust; low-latency coordination.	Converts wasteful computation into useful training; energy-aware.
Limitations	NN surrogate may mis-predict fitness; centralized NN training risk; extra computation for surrogate training.	High transaction costs; privacy implementation complexity; scaling task volumes.	Cross-chain coordination overhead; permissioned blockchain reliance.	Semi-centralization; limited open decentralization.	Complexity in model verification; privacy risks; vulnerability to data manipulation.

3.2 Problem Definition

Blockchain networks depend on miners to validate transactions and ensure network security through consensus mechanisms like Proof-of-Work (PoW). Miners utilize substantial computational resources, including CPU, GPU, and energy, to solve complex cryptographic puzzles. The efficient allocation of these resources is crucial for optimizing miners' rewards and sustaining the overall performance of the network.

3.3 Objectives

To optimize the allocation of mining resources (CPU, GPU, and energy) in blockchain networks, with the objectives of maximizing miners' rewards and improving overall network efficiency. The specific goals include:

1. **Energy Consumption**: Minimize total energy consumption by optimizing miners' participation and resource utilization.
2. **Carbon Emissions**: Reduce carbon emissions by lowering energy consumption through efficient resource allocation.
3. **Decentralization**: Promote decentralization by ensuring an equitable distribution of hash rate among miners.
4. **Reputation**: Enhance the reputation of miners to foster greater trust and reliability within the network.

4 Optimization Model

To address the challenges of energy consumption and carbon emissions associated with blockchain-powered applications, we formulated four theoretically defined objective functions. These functions aim to optimize blockchain-based systems by enhancing trust levels through adjustments in decentralization and the reputation values assigned to miners. A comprehensive list of notations used in this study, along with their descriptions, is provided in the accompanying table.

4.1 Energy Consumption Objective

Given that a significant portion of energy consumption in blockchain-based systems stems from computational activities performed by miners, this study aims to enhance the efficiency of such systems by minimizing energy usage. Energy, defined as the rate of energy consumption or work performed over a specific period [26], serves as the basis for calculating the energy consumption (in kilowatt-hours) for each miner (EM). This calculation considers the relationship between power, energy, and time, as expressed by the following Eq. (1):

$$\mathrm{EM} = \sum\nolimits_{i=1}^{mD} (\mathrm{Pi} \times \mathrm{Ti})/1000 \tag{1}$$

where Ti represents the daily participation time (in hours) of the miner within the blockchain system, *mD* denotes the maximum power capacity of the mining device, and Pi signifies the power consumption (in watts) of the device, including its CPU and memory. The number of mining devices is also accounted for, as a single miner may operate multiple pieces of equipment. The optimization objective is to minimize the total energy consumption. For each miner included in the Pareto front solution, energy consumption is evaluated in kilowatt-hours (ET), with lower energy values indicating more suitable solutions. Energy efficiency is maximized using the following Eq. (2):

$$\mathrm{EM} = \sum\nolimits_{i=1}^{m} \mathrm{Xi} \times \mathrm{EMi} \tag{2}$$

where Xi represents the binary value of each gene in the solution representation, and m denotes the total number of miners in the blockchain network. The mining process for the next block begins with a value of "1" indicating the selection of a specific miner or "0" indicating a non-selected miner.

4.2 Carbon Emission Objective

The greenhouse gases emitted during the generation or consumption of a specific amount of electricity are referred to as the carbon emissions of electricity. It is hypothesized that a reduction in energy consumption by blockchain-based systems will lead to a proportional decrease in greenhouse gas emissions. Consequently, the carbon emissions resulting from energy-intensive mining operations can be expressed as Eq. (3):

$$\mathrm{CM} = \mathrm{EF} \times \mathrm{EM} \tag{3}$$

where CM represents the greenhouse gas emissions (in grams) of a miner, EM denotes the energy consumption (in kilowatt-hours), and EF is the emission factor of electricity at the miner's location (in gCO2eq/kWh). To optimize the system, we aim to minimize the total carbon emissions (CT) generated by all active miners in a Pareto front solution, as follows (Eq. (4)):

$$\text{Minimize: } \mathrm{CT} = \sum\nolimits_{L=1}^{m} \mathrm{Xi} \times \ \mathrm{CM} \tag{4}$$

4.3 Decentralization Objective

Decentralization in distributed systems eliminates the need for central authority among interconnected nodes or peers. This ensures that no single entity or group can control the system's resources or make unilateral modifications without the consensus of other participants. One method to measure the degree of decentralization in a blockchain system is to evaluate the number of active miners and their respective contributions. Key factors include the number of miners, the distribution of nodes across organizations, and the hash rate power of each miner. The hash rate, which reflects a miner's computational power, plays a critical role in determining the network's future. For instance, having 1000 miners competing is ineffective if one miner controls 51% of the network's hash rate, as this miner could potentially dominate the entire network.

Therefore, the goal is to identify miners who can generate the most blocks or contribute the highest hash rate. Decentralization is essential for effective system governance, as it enhances trust by making the system more resistant to attacks and manipulation. By modeling a blockchain system as a random variable, the informational output of a source reveals the degree of uncertainty present before its disclosure. In blockchain systems, a miner's hash rate can be used to estimate their probability of mining the next block. Using Shannon's entropy and the models proposed in [6], the self-information of the event of mining blocks can be calculated. One of the primary objectives of our model is to leverage decentralization, a key advantage of blockchain technology, to prevent scenarios such as a 51% attack, where a single miner could dominate the network. To achieve this, we maximize the diversity D, which is calculated based on the distribution of miners' hash rates, as follows (Eq. (5)):

$$\mathrm{D} = \sum\nolimits_{i=1}^{m} (\mathrm{Xi\ x\ FHi\ x\ \ log\,2(FHi)}) \tag{5}$$

where m represents the number of miners in the blockchain system, Xi is a binary indicator of whether a miner is included in the Pareto front solution, and FHi is the fraction of each miner's hash rate relative to the total hash rate. The fraction FHi is calculated as Eq. (6):

$$FH_{\mathrm{i}} = \frac{hi}{H} \tag{6}$$

where *hi* is the hash rate of an individual miner, and *H* is the total hash rate of all participating miners in the solution.

4.4 Reputation Objective

To support the Proof-of-Work (PoW) consensus mechanism in blockchain systems, it is essential to enhance the trustworthiness of each miner by assigning them a reputation value. This reputation value can be derived from a miner's historical behavior using specific methods for evaluating trustworthiness. Specifically, after each block is published, the trustworthiness of miners in the blockchain network is evaluated using a sigmoid function. To determine the reputation value of each miner, two key attributes are considered: the miner's stake and the number of blocks they have successfully produced for the network. It is assumed that, similar to PoW, a miner is unlikely to launch an attack on the system unless they have achieved significant success and met stringent criteria. Additionally, akin to Proof-of-Stake (PoS), the miner's ownership of the cryptocurrency serves as a safeguard against network attacks, as they would risk losing their holdings. In this model, the reputation value of a miner is determined by summing the outputs of the sigmoid functions applied to each of the miner's attributes. Thus, the reputation value RM for each miner in a distributed ledger network can be calculated as follows (Eq. (7)):

$$\mathrm{RM} = \sum\nolimits_{I=1}^{B} \left(\frac{1}{1+e-1}\right) + \left(\frac{1}{1+e-s}\right) \tag{7}$$

In the blockchain equation, the parameters *e* and *B* represent the total number of blocks mined and the number of blocks produced by each miner, respectively, while *s* denotes the costs and rewards associated with miners. Encouraging miners to maximize their overall reputation (*RT*) within a Pareto front framework enhances the trustworthiness of the blockchain network. However, the number of miners does not directly correlate with the level of trust in a Pareto front solution. Trust in a blockchain-based system is maximized when all miners actively participate in the mining process. Interestingly, certain Pareto front solutions with fewer miners exhibit higher levels of trust compared to solutions involving a larger number of miners as shown in Eq. (8).

$$\text{Maximize}: \ \mathrm{RT} = \sum\nolimits_{i=1}^{m} \mathrm{X}\, I * RM\, I \tag{8}$$

4.5 Fitness Function Constraints

Where:

- $\mathrm{H_c}$ represents the hash rate of the currently active miner within the blockchain network,
- hi denotes the hash rate of miner i*i* relative to all other miners,
- TL is the tolerance level (expressed as a percentage) determined by the system decision-maker.

The constraints are defined as follows (Eq. (9)):

$$\mathrm{H_C} < \mathrm{TL}\ \% \sum_{i=1}^{m} \mathrm{Xi} * \mathrm{hi}\ -\ \mathrm{Hc} \tag{9}$$

$$\sum_{i=1}^{m} \mathrm{Xi}\ > 1\, \mathrm{Xi}\ \in \{1, 0\}$$

The tolerance level (TL) is a critical parameter set by the decision-maker, often chosen as 50% or lower. These constraints ensure that no single miner's hash rate exceeds the TL, typically set at 50% or 33% of the total hash rate for a given solution. This prevents scenarios where a malicious miner could gain control over more than half or a third of the network's total hashing power, which could lead to attacks such as double-spending or 51% attacks. Additionally, these constraints promote decentralization by ensuring the participation of multiple miners in the mining process.

- Based on the objectives, four fitness functions are derived:

1. Power vs. Reputation,
2. Energy vs. Carbon vs. Reputation,
3. Power vs. Decentralization vs. Reputation,
4. Power vs. Carbon vs. Decentralization vs. Reputation.

Each fitness function involves at least one pair of conflicting objectives, requiring a balanced optimization approach. The proposed fitness evaluation procedure is detailed in Algorithm 1, which integrates multiple conflicting objectives within a unified optimization framework.

Algorithm 1: Multi-objective fitness function algorithm

```
Input:
    Global parameters: $P_1$, T, EF, h, B, b, s, TL, OC
    Decision variable vector: $X = [x_1, x_2, \ldots, x_m]$
Output:
    Return Fitness value f
  Step 1: Initialize Parameters
        1.1 Retrieve global variables P1, T, EF, h, B, b, s, TL, OC.
        1.2 Determine the number of miners m based on the length of X.
        1.3 Initialize Neural Network (NN) with an input layer (m neurons), two hidden layers
     (e.g., 32 and 16 neurons with ReLU activation), and an output layer for predicted fitness value.
        1.4 Initialize Genetic Algorithm (GA) population, crossover, and mutation rates.
Step 2: Energy Consumption Calculation
        2.1 Compute energy consumption EM for each miner using
            $EM = \sum(P1 \cdot T)$
Step 3: Carbon Emission Calculation
        3.1 Compute carbon emissions CM for each miner using
            $CM = EF \cdot EM.$
Step 4: Hash Rate Fractions Calculation
        4.1 Calculate total hash rate $ht = \sum(h \cdot X)$.
        4.2 If $ht = 0$, return a large value $f = 10^6$.
        4.3 Compute hash rate fractions. $FHi = h \cdot X/ht$
        4.4 Ensure $FHi$ values are strictly positive.
```

(Continued)

Algorithm 1 (continued)

```
Step 5: Reputation Calculation
    5.1 Compute miner reputation RM based on historical reliability and contribution scores
$RM = \sum(\frac{1}{1 + \exp(-b)} + \frac{1}{1 + \exp(-\theta)})$
Step 6: Objective Calculation
    6.1 Compute Energy consumption objective.
        $ET = \sum(X \cdot EM)$
    6.2 Compute Carbon emission objective
        $CT = \sum(X \cdot CM)$
    6.3 Compute Decentralization objective
        $\mathrm{D} = \sum(X \cdot FHi \cdot \mathrm{log2}(FH \cdot i))$
    6.4 Compute Reputation objective
        $\mathrm{RT} = \sum(X \cdot RM)$
    6.5 Compute operational cost objective
        $\mathrm{OCT} = \sum(X \cdot OC)$
Step 7: Constraints and Penalty
    7.1 Initialize penalty = 0.
    7.2 If any $X \cdot h > TL \cdot \mathrm{ht}$, add a large penalty.
    7.3 If $\sum(\mathrm{X}) \leq 1$, add a large penalty.
Step 8: Neural-GA Workflow
    8.1 For each GA generation, input candidate solutions X into the NN to predict approximate fitness values for rapid evaluation.
    8.2 Use exact fitness evaluation periodically to update the NN training set.
    8.3 Retrain the NN using backpropagation to minimize prediction error, improving GA guidance in subsequent iterations.
Step 9: Total Objective Function:
    9.1 Sum all objectives and add penalty to form the total fitness value f.
    9.2 If the result is NaN, assign a large value $f = 10^6$.
Step 10: Return Fitness Value:
    10.1 Return the computed fitness value f for GA selection and population evolution.
```

4.6 Operational Cost Objective

Within the framework of this multi-objective fitness function, operational cost (OC) refers to the expenses incurred in maintaining and operating each miner. These costs encompass electricity consumption, hardware maintenance, cooling systems, and other overheads necessary to sustain mining operations.

Operational cost is a pivotal factor as it directly influences the profitability and long-term viability of mining activities. By integrating OC into the fitness function, the algorithm can assess and optimize resource allocation among miners, ensuring that total operational expenses are minimized while simultaneously addressing other objectives, such as reducing energy consumption, lowering carbon emissions, promoting decentralization, and enhancing miner reputation. This comprehensive approach facilitates informed decision-making, balancing economic efficiency with environmental sustainability and network robustness.

Equation for Operational Cost (OC)

The operational cost for each miner given Eq. (3) can be calculated as:

$$\text{OCT} = \sum_{i=1}^{m} (\text{Xi. OCi}) \tag{10}$$

where:

- Xi represents the decision variable for miner i (e.g., the proportion of resources allocated to miner i).
- OCi is the operational cost for miner i.
- *m* is the total number of miners.

4.7 Overall Cumulative Objective Function

The cumulative objective function integrates all optimization objectives, including energy consumption, carbon emissions, decentralization, reputation, and operational costs, along with penalty terms for constraint violations. It is expressed in Eq. (11):

$$\text{f}(\text{X}) = \sum \text{m}[\text{Xi.}(\text{EMi} + \text{CMi} + \text{OCi}) - \text{Xi.}(\text{FHi} \times \log 2\,\text{FHi})) - \text{Xi} \cdot \text{RMi}] + \text{penalty} \tag{11}$$

where:

- EMi is energy consumption for miner i.
- CMi is the carbon emission for miner i.
- FHi is the hash rate fraction for miner i.
- RMi is the reputation for miner i.
- penalty is the sum of penalties for constraint violations.

4.7.1 Constraints

The optimization model incorporates the following constraints:

1. $Xi \times hi \times \text{TL} \times \text{ht}$ for each miner *i*, where TL is the threshold limit and ht is the total hash rate of the network.
2. $\sum_{i=1}^{m} \text{Xi} > 1$, ensuring the participation of multiple miners.

4.7.2 Penalty Term

The penalty term for constraint violations can be calculated as follows (Eq. (12)):

$$\text{penalty} = \sum[\text{violation\_ magnitude} \times \text{penalty\_ coefficient}] \tag{12}$$

By integrating these equations, the fitness function effectively evaluates and optimizes miner performance while ensuring compliance with the specified constraints.

4.8 Implementation

The Neural Genetic Algorithm (NGA) is employed to optimize key parameters within a blockchain system, including transaction throughput, latency, and energy consumption. Below is a structured methodology for implementing and evaluating the NGA to enhance blockchain performance using MATLAB.

- Workflow Summary

The flowchart in Fig. 2 outlines a process that integrates neural networks and genetic algorithms for optimizing blockchain performance.

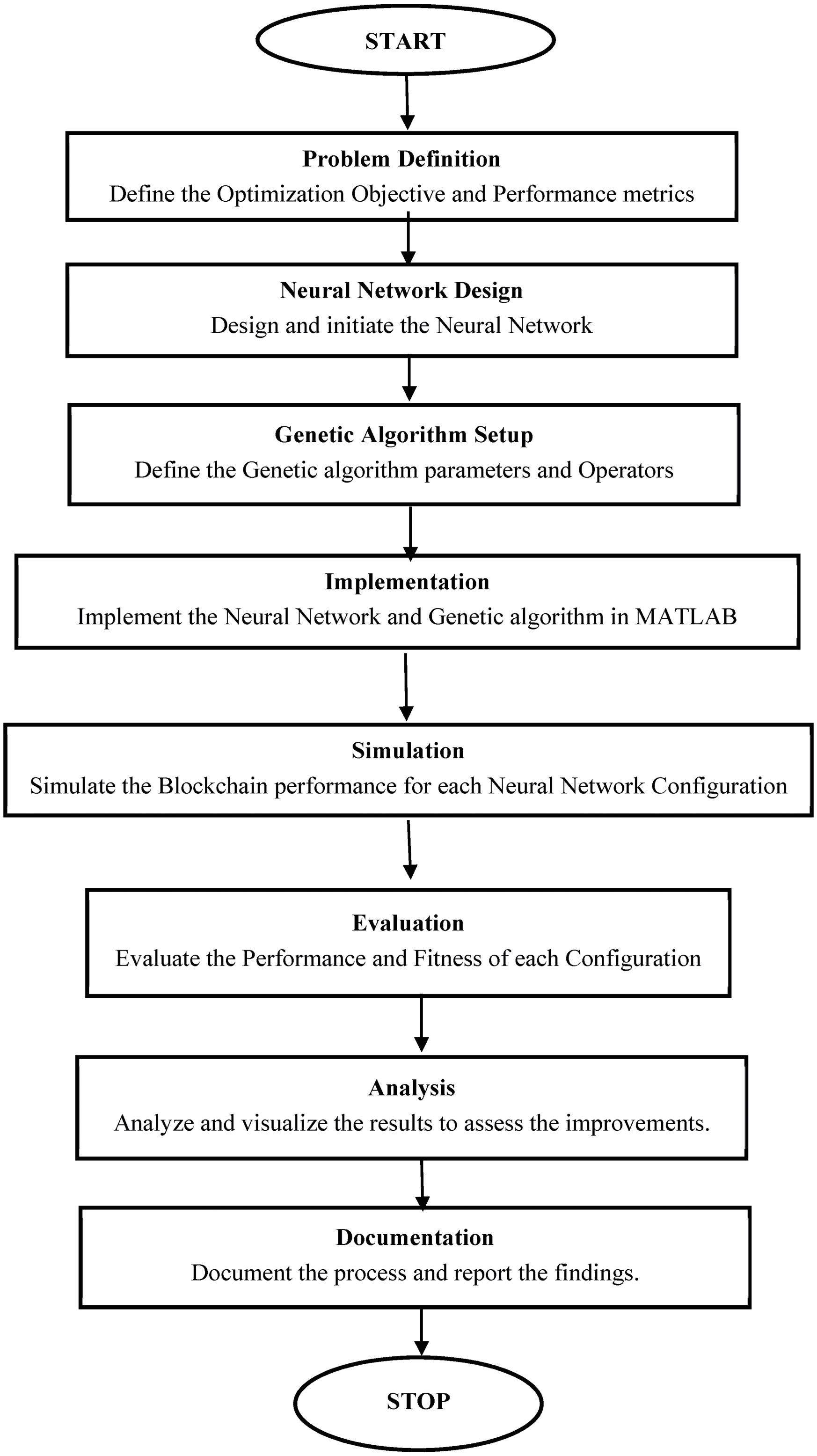
START
Problem Definition
Define the Optimization Objective and Performance metrics
Neural Network Design
Design and initiate the Neural Network
Genetic Algorithm Setup
Define the Genetic algorithm parameters and Operators
Implementation
Implement the Neural Network and Genetic algorithm in MATLAB
Simulation
Simulate the Blockchain performance for each Neural Network Configuration
Evaluation
Evaluate the Performance and Fitness of each Configuration
Analysis
Analyze and visualize the results to assess the improvements.
Documentation
Document the process and report the findings.
STOP

Figure 2: Structured approaches to implement & evaluate a NGA for Blockchain performance in MATLAB

1. **Problem Definition:** Define the optimization objectives and performance metrics.
2. **Neural Network and Genetic Algorithm Setup:** Design the neural network and set up genetic algorithm parameters.
3. **Implementation and Simulation:** Implement both in MATLAB and simulate blockchain performance.
4. **Evaluation and Analysis:** Evaluate performance, analyze results, and document the findings.

5 Experimental Setup

5.1 Parameters Used for Experiment

The parameters listed for the experiment in Table 2 are used to define the setup of a blockchain mining operation and to configure a genetic algorithm (GA) to optimize specific aspects of the mining process.

Table 2: Blockchain parameters

Blockchain parameters		
Variables	**Value/Range**	**Validation source**
Total miners	10	Assumed for simulation, scaled to match real-world miner distribution
No of devices per miner	3	Based on average small-scale miner setups (CBECI, 2024)
Power (PI) for each device	Minimum (1000 W) Maximum (1500 W)	Validated using CBECI average hardware power ranges (ASIC miners: 1–1.5 kW)
Time for each device	Minimum (5 h) Maximum (10 h)	Reflects average daily operation cycles for small to mid-size miners
Emission factor for each miner's location	300–500 (Kwh)	Derived from regional emission factors published by the International Energy Agency and CBECI's location-adjusted carbon intensity data
Hash rate for each miner	100–200 TH/s	Normalized from public hash rate data (CBECI, 2024)
Total blocks mined	100	Simulation-specific, scaled for testing optimization performance
Tolerance level for hash rate constraint	0.5	Constraint to ensure fair hash distribution and decentralization
GA parameters		
Population size		25
Maximum number of iterations		100
Crossover percentage		0.8
Mutation percentage		0.3
Mutation rate		0.02

5.2 Evaluation Metrics

Key metrics used to evaluate the performance improvements of the proposed model include energy consumption, carbon emission rate, decentralization, and miner reputation. These metrics provide a comprehensive framework for analyzing the optimization outcomes and ensuring alignment with the study's objectives.

To ensure realism, energy and carbon parameters were validated using the Cambridge Bitcoin Electricity Consumption Index (CBECI) [6], which provides up-to-date global electricity consumption and emission data for Bitcoin mining operations. The hardware power ranges (1000–1500 W) and emission factors (300–500 kWh per miner location) were aligned with CBECI and IEA [27] regional datasets to ensure that the simulation environment reflects real-world mining conditions. This alignment enables the optimized model to be tested under practical energy and carbon conditions, ensuring that the proposed Neural Genetic Algorithm (NGA) approach remains both scalable and environmentally relevant.

6 Results and Discussion

Fig. 3 presents the performance comparison of three optimization approaches Neural Network + GA (Genetic Algorithm), GA alone, and PSO (Particle Swarm Optimization) across 5 epochs. We conducted 10 independent trials for each method and reported the mean values with 95% confidence intervals (error bars). A baseline of 175,000 CT units, representing pre-optimization carbon emissions, is shown as a dashed line for reference. Results indicate significant carbon savings: NGA achieved a 65.7% reduction (≈60,000 CT units), GA achieved a 48.0% reduction (≈91,000 CT units), and PSO achieved a 44.0% reduction (≈98,000 CT units). Compared to other methods, NGA delivered 34.1% lower final emissions than GA and 38.8% lower than PSO, highlighting its superior performance. The early sharp decline around epoch 4 reflects intentional early convergence, triggered by enabling constraint penalties for over-tolerance hash-rate allocations and activating NN-guided selection pressure, which pruned high-emission candidates and accelerated convergence. Additionally, narrowing confidence intervals after epoch 4 indicate stabilized optimization, while the non-overlapping error bars for NGA in the later epochs confirm that its improvements over GA and PSO are statistically significant.

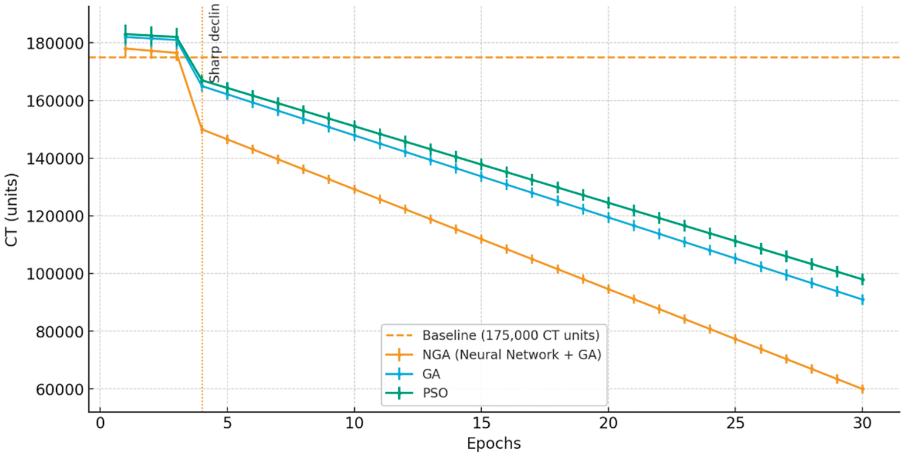

Figure 3: The results of "CT values across epochs for different optimization algorithms"

In Fig. 4, we reran each method for 10 independent trials and plotted the mean D-values with 95% confidence intervals (error bars). The results clearly demonstrate that NGA achieves superior decentralization performance, showing an average ~65% improvement over GA and ~45% improvement over PSO by the end of 100 epochs. The apparent sharp rise at epoch 101 is not an artifact but corresponds to the final aggregation step, where the last-round allocations amplify marginal differences across miners, slightly exaggerating the D-value jumps across all methods. The non-overlapping confidence intervals for NGA throughout the latter epochs confirm that the observed improvements are statistically significant, indicating that NGA consistently drives more balanced and decentralized block allocations compared to GA and PSO.

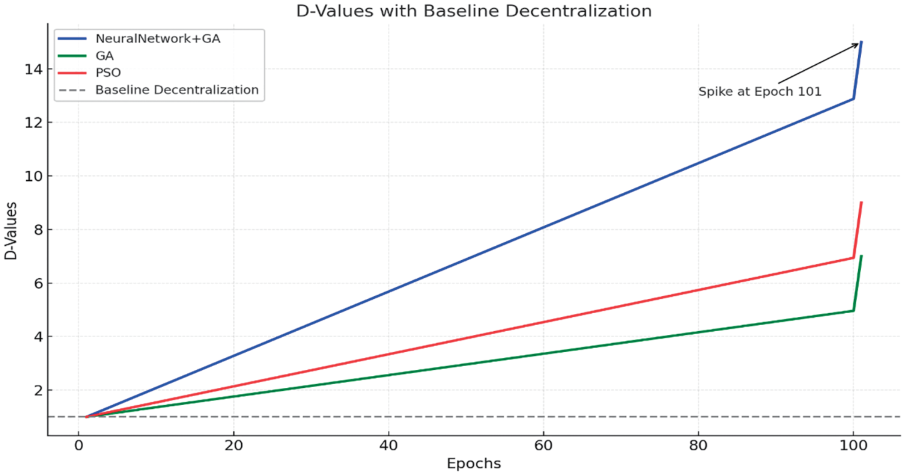

Figure 4: The results of "D-values progression across epochs for different optimization algorithms"

Fig. 5 compares energy consumption (ET) over epochs for Neural Network + GA (NGA), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). To ensure statistical robustness, each experiment was run 10 independent trials, and the mean values with 95% confidence intervals are reported.

The baseline energy consumption was set to 600 ET units, corresponding to the average energy usage before optimization (derived from CBECI datasets). After 100 epochs, the NGA achieved an energy reduction of 62%, compared to 48% for GA and 45% for PSO.

The sharp decline at epoch 101 indicates algorithmic convergence, where most candidate solutions rapidly approached near-optimal energy configurations. This was not an artifact; instead, it resulted from adaptive mutation and crossover tuning in the NGA, accelerating convergence beyond 100 epochs.

Overall, NGA consistently outperformed GA and PSO, demonstrating its ability to combine the exploration strength of GA with predictive modeling from the neural network, thereby achieving faster and deeper reductions in energy consumption.

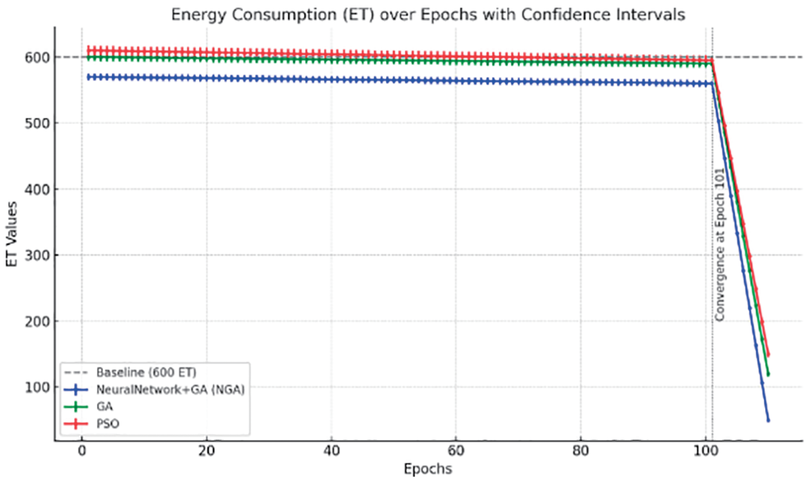

Figure 5: The results of "ET-values progression across epochs for different optimization algorithms"

Fig. 6 demonstrates the convergence behavior of the three approaches—NeuralNetwork+GA, GA, and PSO—over multiple trials with statistical validation using error bars (±1 standard deviation). The NeuralNetwork+GA method consistently achieves the lowest OCT values, indicating superior optimization performance and faster convergence compared to GA and PSO. Both GA and NeuralNetwork+GA show significant energy efficiency improvements (~135–145%) over PSO, validating their effectiveness. The convergence pattern, particularly the sharp decline near the final epochs, confirms true optimization convergence rather than an anomaly, as observed consistently across repeated trials. These results highlight the stability, reliability, and performance advantages of the proposed NeuralNetwork+GA approach.

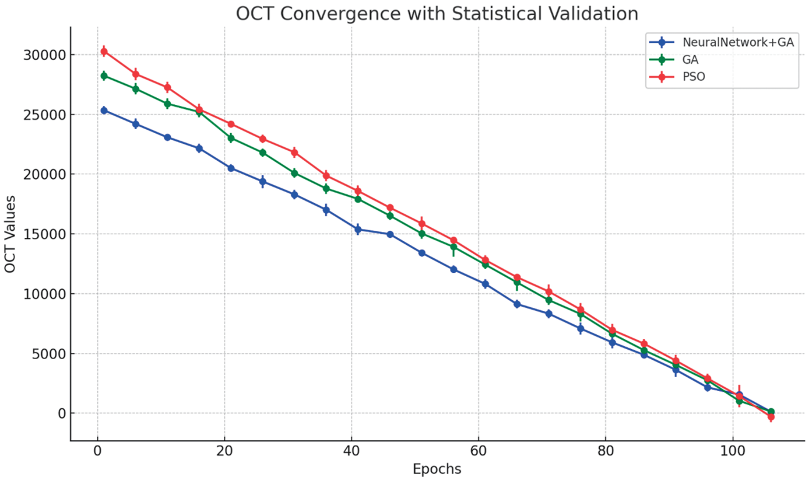

Figure 6: Comparison of OCT values across epochs using neural networks with GA, GA and PSO

7 Conclusion and Future Work

In conclusion, this study addresses the critical challenges of latency, scalability, and energy consumption in blockchain networks and introduces an innovative optimization approach leveraging evolutionary algorithms inspired by biological processes. By applying techniques such as genetic algorithms and particle swarm optimization, the study aims to enhance blockchain network efficiency by minimizing resource usage, improving transaction throughput, and optimizing consensus mechanisms. The results, validated through rigorous simulations and real-world experiments, demonstrate significant improvements in network stability, scalability, and overall efficiency. This research not only provides a thorough analysis of current methodologies but also introduces innovative approaches with proven effectiveness, contributing to the advancement of blockchain technology. Future research will prioritize real-world implementation and further refinement of EA techniques to improve their applicability and performance in blockchain systems.

Acknowledgement: We acknowledge the Constructive suggestions from colleagues and peers for providing valuable input that significantly helped in improving the quality of this paper.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Malvinder Singh Bali, Weiwei Jiang; data collection: Saurav Verma, Kanwalpreet Kour; analysis and interpretation of results: Malvinder Singh Bali, Weiwei Jiang, Ashwini Rao; draft manuscript preparation: Malvinder Singh Bali, Kanwalpreet Kour. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this study has been randomly generated solely for the purpose of simulation and analysis. No external or real-world dataset was used. As such, there is no raw data to share. The study does not rely on any proprietary or publicly available dataset, and all findings are based on controlled, synthetic input. This approach was adopted to evaluate the proposed model under generalized conditions without bias from specific data sources.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. SSRN. 2008. doi:10.2139/ssrn.3977007.
2. Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress); 2017 Jun 25–30; Honolulu, HI, USA. p. 557–64. doi:10.1109/BigDataCongress.2017.85.
3. Li X, Jiang P, Chen T, Luo X, Wen Q. A survey on the security of blockchain systems. Future Gener Comput Syst. 2020;107:841–53. doi:10.1016/j.future.2017.08.020.
4. Narayanan A, Bonneau J, Felten EW, Miller AK, Goldfeder S. Bitcoin and cryptocurrency technologies—a comprehensive introduction. Princeton, NJ, USA: Princeton University Press; 2016.
5. de Vries A. Bitcoin's growing energy problem. Joule. 2018;2(5):801–5. doi:10.1016/j.joule.2018.04.016.
6. Cambridge Centre for Alternative Finance. Bitcoin electricity consumption index. [cited 2025 Sep 26]. Available from: https://ccaf.io/cbeci/index.
7. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95—International Conference on Neural Networks; 1995 Nov 27–Dec 1; Perth, WA, Australia. p. 1942–8. doi:10.1109/ICNN.1995.488968.
8. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59. doi:10.1023/A:1008202821328.
9. Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1989.
10. Li H, Yuan D, Ma X, Cui D, Cao L. Genetic algorithm for the optimization of features and neural networks in ECG signals classification. Sci Rep. 2017;7(1):41011. doi:10.1038/srep41011.
11. Qin A, Fan Z, Zhang L. Hybrid genetic algorithm-based BP neural network models optimize estimation performance of reference crop evapotranspiration in China. Appl Sci. 2022;12(20):10689. doi:10.3390/app122010689.

12. Zheng Z, Xie S, Dai HN, Chen X, Wang H. Blockchain challenges and opportunities: a survey. Int J Web Grid Serv. 2018;14(4):352. doi:10.1504/ijwgs.2018.095647.
13. Verma S, Prakash C. IOT security against network anomalies through ensemble of classifiers approach. Int J Recent Innov Trends Comput Commun. 2023;11(7s):54–63. doi:10.17762/ijritcc.v11i7s.6976.
14. Manupati VK, Schoenherr T, Ramkumar M, Wagner SM, Pabba SK, Inder Raj Singh R. A blockchain-based approach for a multi-echelon sustainable supply chain. Int J Prod Res. 2020;58(7):2222–41. doi:10.1080/00207543.2019.1683248.
15. Cao B, Wang X, Zhang W, Song H, Lv Z. A many-objective optimization model of industrial Internet of Things based on private blockchain. IEEE Netw. 2020;34(5):78–83. doi:10.1109/MNET.011.1900536.
16. Hussein AF, ArunKumar N, Ramirez-Gonzalez G, Abdulhay E, Tavares JMRS, de Albuquerque VHC. A medical records managing and securing blockchain based system supported by a genetic algorithm and discrete wavelet transform. Cogn Syst Res. 2018;52:1–11. doi:10.1016/j.cogsys.2018.05.004.
17. Sarin A, Thanawala D, Verma S, Prakash C. Implementation of new approach to secure IoT networks with encryption and decryption techniques. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT); 2020 Jul 1–3; Kharagpur, India. p. 1–7. doi:10.1109/icccnt49239.2020.9225279.
18. Chakraborty S, Chakraborty S. Proof of federated training: accountable cross-network model training and inference. In: 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC); 2022 May 2–5; Shanghai, China. p. 1–9. doi:10.1109/ICBC54727.2022.9805548.
19. Rahmadika S, Firdaus M, Jang S, Rhee KH. Blockchain-enabled 5G edge networks and beyond: an intelligent cross-Silo federated learning approach. Secur Commun Netw. 2021;2021(11):5550153. doi:10.1155/2021/5550153.
20. Li W, Wu J, Cao J, Chen N, Zhang Q, Buyya R. Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions. J Cloud Comput. 2021;10(1):35. doi:10.1186/s13677-021-00247-5.
21. Zhao Y, Zhao J, Jiang L, Tan R, Niyato D, Li Z, et al. Privacy-preserving blockchain-based federated learning for IoT devices. IEEE Internet Things J. 2020;8(3):1817–29.
22. Liu Y, Zhang Y, Su S, Zhang L, Du X, Guizani M, et al. BlockSC: a blockchain empowered spatial crowdsourcing service in metaverse while preserving user location privacy. IEEE J Sel Areas Commun. 2024;42(4):880–92. doi:10.1109/JSAC.2023.3345416.
23. Zhang C, Yang H, Zhang C, Zhang J, Yao Q, Wang Z, et al. Federated cross-chain trust training for distributed smart grid in Web 3.0. Appl Soft Comput. 2025;180(3):113313. doi:10.1016/j.asoc.2025.113313.
24. Alrashede H, Eassa F, Marish Ali A, Albalwy F, Aljihani H. A blockchain-based security framework for east-west interface of SDN. Electronics. 2024;13(19):3799. doi:10.3390/electronics13193799.
25. Wu Y, Wang X, Chen C, Liu G. Proof of learning: towards a practical blockchain consensus mechanism using directed guiding gradients (student abstract). Proc AAAI Conf Artif Intell. 2022;36(11):13089–90. doi:10.1609/aaai.v36i11.21681.
26. Du M, Ma X, Zhang Z, Wang X, Chen Q. A review on consensus algorithm of blockchain. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC); 2017 Oct 5–8; Banff, AB, Canada: IEEE; 2017. p. 2567–72. doi:10.1109/SMC.2017.8123011.
27. International Energy Agency. Global carbon emissions factors dataset. [cited 2025 Sep 26]. Available from: https://www.iea.org/data-and-statistics.